

Laborübung 1 Rechnerarchitektur

Hinweise: Schreiben Sie die Lösungen, so weit es möglich ist, auf die Aufgabenblätter. Tragen Sie Name, Matrikelnummer und Studiengang in die nachfolgende Tabelle ein und schreiben Sie auf jedes zusätzlich abgegebene Blatt ihre Matrikelnummer. Lassen Sie vorgeführte Experimente vom Betreuer gegenzeichnen. Für eine Bescheinigung der erfolgreichen Teilnahme sind in jeder bis auf eine Laborübung mindestens 40% und insgesamt mindestens 50% der Punkte erforderlich.

Name	Matrikelnummer	Studiengang	Punkte	$\geq 40\%$

Aufgabe 1: Test des Einführungsprogramms. Dem Betreuer sind vorzuführen:

- Test im Schrittbetrieb 0,5P
- Test mit Unterbrechungspunkten 0,5P
- Test mit Schaltereingaben und 0,5P
- Test mit Änderung von Registerinhalten. 0,5P

Punkte Aufgabe 1	
------------------	--

Aufgabe 2: Legen Sie ein neues Projekt Logik1 an. Löschen Sie Logik1.c

- View > Solution Explorer (Strg+Alt+L)
- Rechtsklick auf die Datei > Remove > Delete

legen Sie eine Assemblerdatei Logik1.s an

- Project > Add New Item (Strg+Shif+A)> Assembly File, Name: Logik1.s

Geben Sie den Programmtext aus Abb. 1 ein.

- Lassen Sie sich vom Betreuer einen Einabewert vorgeben. Ergänzen Sie im Programmkommentar auf dem Aufgabenblatt für jede Zeile, welchen Wert die Anweisung dem Zielregister zuweist. 4P
- Bestimmen Sie experimentell für alle Variationen der Schalterstellungen die Ausgabe auf LD0 bis LD3 und füllen Sie die Wertetabelle in Abb. 1 aus. 4P
- Bilden Sie die Funktion der Schaltung durch logische UND- und ODER-Gatter nach und zeichnen Sie das Ergebnis in Abb. 1 unten ein. (Sie benötigen mindestens 3 Gatter.) 2P

Punkte Aufgabe 2	
------------------	--

```

9   .global main
10  main:
11  ser r16          ; r16: 0b.... ....
12  out 7, r16      ;DDRC: 0b.... ....
13  loop:
14  in r16, 0       ; r17: 0b.... ....
15  mov r17, r16    ; r17: 0b.... ....
16  andi r17, 0b01  ; r17: 0b.... ....
17  mov r18, r17    ; r18: 0b.... ....
18  lsl r17         ; r17: 0b.... ....
19  and r17, r16    ; r17: 0b.... ....
20  or r18, r17     ; r18: 0b.... ....
21  lsl r17         ; r17: 0b.... ....
22  and r17, r16    ; r17: 0b.... ....
23  or r18, r17     ; r18: 0b.... ....
24  lsl r17         ; r17: 0b.... ....
25  and r17, r16    ; r17: 0b.... ....
26  or r18, r17     ; r18: 0b.... ....
27  out 8, r18     ;PORTC: 0b.... ....
28  rjmp loop

```

																Punkte	
SW4	SW3	SW2	SW1	LD3	LD2	LD1	LD0	SW4	SW3	SW2	SW1	LD3	LD2	LD1	LD0		
0	0	0	0					1	0	0	0						
0	0	0	1					1	0	0	1						
0	0	1	0					1	0	1	0						
0	0	1	1					1	0	1	1						
0	1	0	0					1	1	0	0						
0	1	0	1					1	1	0	1						
0	1	1	0					1	1	1	0						
0	1	1	1					1	1	1	1						

Punkte



Punkte

Abbildung 1: Programm, Wertetabelle und Zeichenfläche für die Schaltung zu Aufgaben 2

Aufgabe 3: Legen Sie ein neues Projekt AddSW an. Löschen Sie auch hier die automatisch generierte Datei mit main() und fügen Sie eine Assembler-Datei zum Projekt hinzu. Übernehmen Sie die Zeilen 9 bis 13 aus Abb. 1 als Anfangszeilen und den Sprung auf Zeile 28 als Programmabschluss. Schreiben Sie ein Programm, das die Summe der Schalterwerte SW1 bis SW4 entsprechend

SW4	SW3	SW2	SW1	LD2	LD1	LD0	*	SW4	SW3	SW2	SW1	LD2	LD1	LD0	*
0	0	0	0	0	0	0		1	0	0	0	0	0	1	
0	0	0	1	0	0	1		1	0	0	1	0	1	0	
0	0	1	0	0	0	1		1	0	1	0	0	1	0	
0	0	1	1	0	1	0		1	0	1	1	0	1	1	
0	1	0	0	0	0	1		1	1	0	0	0	1	0	
0	1	0	1	0	1	0		1	1	0	1	0	1	1	
0	1	1	0	0	1	0		1	1	1	0	0	1	1	
0	1	1	1	0	1	1		1	1	1	1	1	0	0	

Der Betreuer lässt sich das Programm mit vier zufälligen Eingaben vorführen und trägt in die zugehörigen Tabellenzeilen in die Spalte * 1 ein, wenn die Ausgabe korrekt ist und sonst -1 (ein Punkt Abzug).

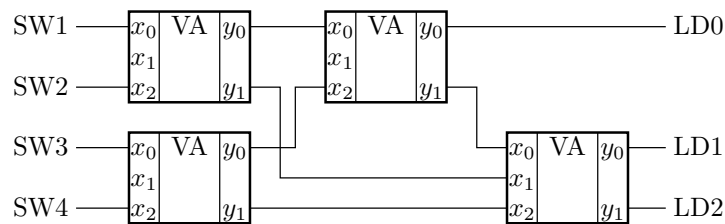


Abbildung 2: Wertetabelle und Schaltungsvorschlag zu Aufgabe

der Tabelle in Abb. 2 bildet und auf die Leuchtdioden LD2 bis LD0 auf der Ansteckbaugruppe ausgibt.

Hinweis: Die Addition von 4 Bits ist nachbildbar mit drei Addierern für zwei Eingabebits (Halbad-dierer) und einen Addierer für drei Eingabebits (Volladdierer) (Abb. 2 unten). Schaltungsbeispiele für Halb- und Volladdierer finden Sie mit google.

Punktevergabe:

-4...8P

- Sinnvolles Programm. 2P
- Programm ist auch übersetz- und ausführbar. weitere 2P
- Für jedes der vier Tests, einen zusätzlichen Punkt, wenn die Ausgabe richtig und einen Punkt Abzug, wenn die Ausgabe falsch ist. -4P bis 4P

Punkte Aufgabe 3	
------------------	--